Icon Editor. (V 1.1)

Load Icon Image Data

Enter Icon Name
(.info Will Be Added)

df0:trashcan

Load Icon Image

Cancel The Load

Next Meeting
Sunday, August 10th at 2pm
Guest Speaker
Tony Cuffe of Commodore

AUG Public Domain Disks

Somehow, we seem to have missed out on describing a few volumes of the public domain library disks between the first and second issues of Workbench. Here are the disks that haven't already been covered.

AUG Library Disk 12

amiga3d — Shows a rotating 3 dimensional solid "Amiga sign". No source available, executable only (downloaded from a Denver BBS) Author: Barry (bart) Whitebook of C-A

ArgoTerm — Yet another terminal emulator program. However, this one is written in macro assembler, so should be of interest to anyone doing serious assembly language development. Version 0.20 Author: Jez San

arrow3d — Shows a rotating 3 dimensional wire frame arrow. No source available, executable only (downloaded from a Denver BBS) Author: Steve Beats of C-A

ld4 — Another directory listing program using some graphics style output. Executable only, no source available. (Downloaded from a Denver BBS)

IconExec & SetWindow — These two tools allow execution of a program from an icon without having to recompile the program. Author: John Toebes VIII

images — Miscellaneous DPaint and digitized pictures, in iff format. Some are rated R.

SetAlternate — Merge the images from two icons to produce one icon with a primary image, and a possibly completely different image to display when selected. Author: John Toebes VIII

StarTerm — Another terminal emulator program. ASCII and XMODEM support, telephone dialer, function keys, load file stripping, text file conversion, full/half duplex. Author: Jim Nangano and Steve Plegge

AUG Library Disk 14

amiga3d — Shows a rotating 3 dimensional solid "AMIGA" sign. This is an updated version of the program released on disk number 12, and now includes full source. Author: Barry Whitebook @ Commodore-Amiga

beep — Source for a function that generates a beep sound, like CTRL-G on a VT100 terminal. Author: Samuel Dicker @ Commodore-Amiga

dex — Program to extract documentation in a human readable format inside source files, and produce nroff style output for manuals and other such external documents. First ever public release. Author: Fred Fish

dimensions — Programs to demonstrate three and four dimensional graphics. Not quite sure how else to describe them! Author: Anselm Hook

filezap — An updated version of the file zap utility first released on disk number 10. Can be used to patch any type of file. Nice, and VERY useful. Author: John Hodgson

gfxmem — An updated version of the graphical memory display program first released on disk number 1. Watch your machine's memory usage change dynamically under use! Author: Louis Mamakos

gi — Converts DPaint brush files to C source files "necessary to create an Image structure, including height, width, depth, and color information, as well as the array of data which represents the bit planes of the image". Author: Mike Farren

pdterm — A simple terminal emulator that does ANSI or DEC VT-100 emulation in 80 cols by 25 lines. Version 1.21. Author: Michael McInerny

shell — A simple csh style shell with history and some other goodies. Still needs some polishing and enhancement, but is quite nice as it is. Thanks Matt!!! We've really needed something like this for a long time. Now if you would just do a ksh version instead... Author: Matt Dillon

termcap — A (mostly) unix compatible implementation of a termcap library. First ever public release. Author: Fred Fish

AUG Library Disk 16

"IFF" - Interchange File Format

The files contained on this disk are the text and sample code for producing and using the Interface File Format developed jointly by Electronic Arts and Commodore-Amiga, Inc.

We at Commodore-Amiga have adopted this standard for internal use and we encourage our developers to do so as well. The standard allows programs developed independently to easily share data with one another.

Electronic Arts has placed all of this sample code into the public domain to encourage others to adopt the standard and to use it.
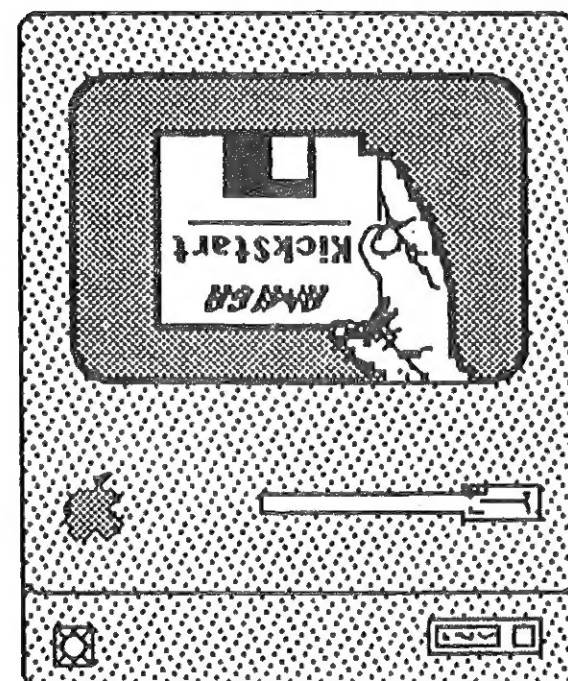
A mock up of the new Mac startup screen?

AMIGA (tm) TECHNICAL NOTE   29-Mar-86

Upgrading an AMIGA A1000 with a Motorola MC68010L8

by Thad Floryan

For quite a while now, I've been experimenting using an MC68010L8 in place of the MC68000 processor in my Amiga. The benefits of the MC68010 are great, but several "problems" with the MC68010 upgrade have held me back from recommending this simple enhancement to everyone. Now, the final quirk has been solved, and I am recommending the upgrade to everyone!

This document has several parts, ALL of which must be read and understood by anyone wishing to perform the upgrade or contract for someone to perform the upgrade for you. Additionally, there are several files which accompany this technical note:

```
CHOP.BAS
CHOP.C
DeciGEL
DeciGEL.asm
DeciGEL.make
```

The several parts of this document are:

I. The MC68010 hardware upgrade

II. The MC68010 software upgrade

III. Copy of Motorola's MC68000 Micro Minutes MM-444-02, entitled "Advantages of Upgrading an MC68000 to an MC68010".

IV. Side notes

I. The MC68010 hardware upgrade

The MC68010 is pin-for-pin compatible with the MC68000 installed in all Amiga Model A1000 PCs. The hardware upgrade consists of replacing the MC68000 with the MC68010.

** CAUTION **

The internals of the Amiga are sensitive to static and are fragile; working on a computer is not like working on your car. Also, opening the Amiga will void your warranty. It may be worth your peace of mind to contract an experienced technician for the hardware part of this upgrade. Common sense and a good mechanical aptitude are essential. Take your time and do not rush; restrain your excitement for later, after the MC68010 is installed.

To open the Amiga, the five screws on the underside of the case must be removed. Releasing the cover can be tricky; examine the left side of the case near the power switch (front side corner) and also 10 inches further back along the seam and note the tabs: push in until they separate, then lift the cover up toward the right side of the case and release the tabs there. The metal RFI shield covers the entire PC board and must be removed to gain access to the MC68000. Remove all the screws along its edge, and also straighten out the tabs which protrude upwards through the shield, then lift and remove the shield; be sure to note where each screw is supposed to be located as there are several types of screws. The disk drive must also be removed since it sits over one edge of the MC68000. First unplug the two cables (power and signal) after noting the orientation so that they can be replaced correctly later. Now remove all screws fastening the disk drive and lift it up and out: NOTE that the wires to the LED are still fastened; these do not need to be removed if you turn not be twisted back into place.

Reconnect all the cables to the Amiga. Powerup the Amiga and boot with your usual Kickstart and Workbench disks. Verify that the system appears to be working as normal. Now, either from the Workbench or CLI, startup the CALCULATOR tool, and perform the operation: " 9 * 3 = ". There should be a software failure; this indicates the MC68010 is running correctly because the CALCULATOR supplied with V1.1 uses a "MOVE SR,ea" instruction which is invalid in user-mode with a MC68010. Reboot the Workbench and proceed now to section II, software upgrade.

If you do not experience the symptoms as described above, there may be several reasons:

1. You re-installed the MC68000 instead of the MC68010,

2. The MC68010 is "dead", or

3. Something else inside the Amiga has been damaged.

At this point, you are on your own. My advice is to re-install the MC68000 and verify that your system works again; if it does, then your MC68010 chip may be bad. If your system still does not operate with the MC68000 back in, then you gotta heap of trouble. Re-read the DISCLAIMER above, and seek competent technical service.

II. The MC68010 software upgrade

Beginning with V1.1 of Kickstart and Workbench, Commodore-Amiga "officially" supports operation of the MC68000, MC68010 and MC68020 CPU chips in the Amiga Model A1000 per:

> "With the exception of the Calculator, software in the 1.1 release is compatible with the 68010 and 68020 processors. Application programmers wishing to take advantage of such products should be sure to use the Exec GetCC() function to get the processor condition codes. (GetCC() handles the differences between the processors such as the fact that "MOVE SR," is an invalid user mode operation in the 68010 and 68020.) In addition, programmers should be sure to NOT use the upper 8 bits of a pointer for storing unrelated information, since the 68020 uses all 32 bits for addressing.
>
> * For those doing systems work, Exec maintains flags in the AttnFlags field of ExecBase which describe the type of processor in your machine. Exec's coldstart procedure will update certain library entry vectors as neccessry to maintain compatibility. If you write code that uses the Supervisor() function, keep in mind that your stack frame is processor dependent. Use the SuperState() function for processor independent supervisor mode entry.
>
> * V1.1 ROMWack will correctly display information for 68010/68020 address and bus errors."

OK, as you can see, simply by having installed the MC68010, you will be able to use most Amiga system software without any problems. But, the CALCULATOR and other programs loaded with libraries using "MOVE SR,ea" will cause your system to go to lunch with the Guru. This is the reason I haven't encouraged people to upgrade their Amigas until now; only a knowledgeable person would have been able to either live with the occasional problem, or rebuild the software to not use the "MOVE SR,ea" instructions.

Write about your AMIGA experiences
Share them with other AMIGA Users Group members

So what happened all of a sudden to change my mind? Simply that one clever person has developed a short and elegant software solution that prevents the "MOVE SR,ea" instructions from crashing your system. "How," you may ask, "is this done?". It's done by inserting a "wedge" that intercepts instruction privilege violations, examining the offending instruction for the presence of "MOVE SR,ea" and, if found, replacing that instruction in memory with a "MOVE CCR,ea" and resuming at the point of failure. Since the only valid use of the system status register (SR) in a user-mode program would be to get the condition codes, the assumption made by the "wedge" is a reasonable one. And, it works! I've been testing ALL the cases that failed previously with the MC68010 in the Amiga and they ALL perform correctly now.

I have included 5 programs with this technical note:

CHOP.BAS	public domain, author: Anonymous
CHOP.C	public domain, author: J S Plegge
DeciGEL	public domain, author: Scott Turner
DeciGEL.asm	public domain, author: Scott Turner
DeciGEL.make	public domain, author: Scott Turner

DeciGEL is the "wedge" program that needs to be placed in the "c" directory of your workbench disk; DeciGEL.asm is the Assembler source file, and the DeciGEL.make is an EXECUTE file that will recreate DeciGEL.asm. If you don't have the Amiga Assembler, you can simply download DeciGEL and use it immediately. It is best to include a line in "s/startup-sequence" on your Workbench disk that will call DeciGEL. For example:

```
echo "Workbench disk. Release 1.1"
echo " "
echo "Use Preferences tool to set date"
echo " "
DeciGEL
.
.
```

The CHOP.BAS and CHOP.C programs are used if you download DeciGEL and use the progam directly, without rebuilding. The CHOP programs must be used to make SURE the DeciGEL program is exactly 168 bytes long. You can determine the size of DeciGEL on your system by using the CLI's LIST program. If the size is greater than 168 bytes, CHOP will allow you to truncate (or chop) the file to 168 bytes. CHOP.C operates with ABasiC (not AmigaBASIC), and CHOP.C is the source of a "C" program; use either one as appropriate for your needs.

With the MC68010 installed and DeciGEL "wedged", your Amiga should operate anywhere from 4% to 50% faster depending on what you're doing. Note that the "Boing!" demo will not be faster since its timing is synchronized to the vertical framing of the monitor's clocking which is FIXED per NTSC TV standards! The biggest benefits of the MC68010 upgrade will occur with heavy number-crunching applications, such as spread sheets and Mandelbrot picture generations. Enjoy!

TextCraft Review

I first heard about Textcraft in a software release from the store where I bought my Amiga, but the one page I got was not terribly descriptive and, further, I didn't believe most of it.

When I saw Textcraft demonstrated (to someone else - I looked over their shoulder) it wasn't terribly impressive. The colours on screen looked awful and amateurish. I learnt later that a previous customer had played havoc with the Preferences page and changed all the colours. Worse, the person I saw actually playing with the program couldn't use the mouse properly and had no idea of how to do anything. Not a good start.

So, after this inauspicious beginning what did I do? Did I ask clever and penetrating questions of the dealer? Did I try things out for myself? No and no. I did what everybody else probably does. I thought about it for half a day, then ran into the shop and said, "Gimme a Textcraft and hurry, I'm double parked!" And they did. Did the dealer question my needs or requirements? Go read science fiction!

When I got back to the office (yes, I'm a BU - Business User) I opened the package. One disk (make a backup STRAIGHT AWAY they said), one manual, and about a million pages of corrections to the manual. The corrections, to be fair, mostly concern themselves with the fact that almost all illustrations of the mouse show the wrong buttons and there are additional features on disk which are not described in the manual.

USING IT:

Textcraft replaces Workbench as far as loading is concerned. In fact, it does not run too happily on top of Workbench. I have found it best to re-boot when using Textcraft.

Once booted, double-click on the disk icon, double-click on the Textcraft icon (there are both 256K and 512K versions. Trash one before you start for more disk space) you are presented with a screen which has a menu bar, a ruler (0-70) inside which are symbols for various page formats (left, centre, right, block) and single or double space symbols. There are other symbols to move the cursor (pencil), delete text (scissors), remember text (camera), reformat the screen and font styles. Page number and document title are displayed and there are five pull-down menus (a bit like Notepad) each with keyboard shortcuts.

There is also a 'Help' menu with quick-reference pages but, best of all, there are some twenty '1-minute Tutorials' which, once selected, actually take over and the computer 'animates' the cursor and other movements necessary to do whatever the Tutor wants to show you. Each Tutorial concentrates on one topic only. You can enter and exit the Tutor without disrupting your document.

ACTUALLY USING IT:

No problem. You could literally begin typing a document within minutes. Also, there are ready-made forms which automatically format memos, business letters, resumes, reports and term papers. These forms are fine, but you cannot get back to the format pages. That is, if you select Business Letter, you cannot later change the name and address of the recipient except on the document itself.

There is absolutely no mailmerge function and if you want to send the same letter to different people, Textcraft will not do it for you. However, for casual use of word-processing, Textcraft is great. I use it to write memos, draft letters and as a tool to organize documents (like letters begging for money. Do you beg first and then plead, or plead first and then beg? Textcraft will swap them around until you're happy).

CONCLUSION:

Don't buy Textcraft if you're a professional writer. Don't buy it to replace the typing pool. DO buy it if you need to write letters occasionally or need to organize your ideas before you commit them to print, or just to file information in document form.

By comparison, our other office machine (well-known US brand with 10Mb hard disk) was running WordStar after-hours for training and the person using it couldn't find her file, or remember how to address the disk and kept looking in the manual, which is the size of a blockbuster novel. She had to give up after several frustrating hours. Mind you, WordStar is considered a good program and it is; it's just so hard to learn. Textcraft doesn't have to be learned, just used.

- John Holland

Build Your Own External Disk Drive

This article is aimed at Amiga owners who have some knowledge of digital electronics and a fair degree of skill with a soldering iron.

Some of you may have seen my machine at the last meeting running a pair of five inch external floppy drives. This article describes the modifications I made to the drives to make them work with the Amiga. These are NOT the same mods published recently in Amazing Computing; unlike the Amazing mods, which were designed to connect a 40 track IBM-compatible drive (360K) to work with the Transformer IBM emulator, these circuits allow you to use an 80 track double sided drive with AmigaDOS and get exactly the same 880K per disk as the 3.5 inch drives. The main advantage of the five inch drives is that disks are CHEAP - about a quarter of the cost of 3.5 inch. The drives themselves are also comparatively inexpensive. You should be able to pick up a suitable pair of five inch drives for less than $200 each, and a case and power supply for well under $100. Add a little time and effort, and you've got twice as much storage as a standard external drive for much less money.

Those of you who simply must have all your disks the same size, don't feel left out. These circuits can also be used to adapt any industry standard double sided 3.5 inch drive for use with the Amiga.

The Amiga's external drive port is nonstandard in several ways. Not as utterly nonstandard as those of the Apple //e or the Macintosh, but enough to stop you having any success with an unmodified drive. Firstly, the connector on the back of the machine is a DB23 socket, for which matching plugs are virtually impossible to get. Secondly, a few of the drive control signals are used in nonstandard ways (see Appendix E of the hardware reference manual for a full explanation).

Fig. 1 shows the circuitry needed to adapt a standard floppy drive for use with the Amiga. If you would rather make up an external adaptor board than solder extra bits and pieces onto your disk drive, you will also need the extra parts shown in Fig. 5. In either case, you will need a separate set of Motor On and Disk Changed latches (Fig. 1) for each external drive.

The interface signals shown in Fig. 1 are as follows:

DRESB*   is an output from the Amiga (pin 10 of the DB23) used to reset the drive on power up. After asserting DRESB*, the Amiga examines the READY* output of the drive to find out what kind of drive it is dealing with. The circuit in Fig. 1 identifies the drive as an 80 track, double sided type.

MTRXD*   is the Amiga's MOTOR ON signal for external drives (pin 8 of the DB23). The Amiga sets up MTRXD* before selecting a drive, and expects the drive to latch it when selected. This gives the Amiga independent control over each external drive's spindle motor.

If you build an external adaptor board, MTRXD* will need a 1k pullup resistor (see Fig. 5). If you modify your drive and use the cable in Fig. 2, MTRXD* will be pulled up by the resistor normally attached to the drive's MOTOR ON* line.

SEL   is the inversion of one of the Amiga outputs SEL1B*, SEL2B* or SEL3B* (DB23 pins 21, 9 and 20 respectively). It goes high when the appropriate drive is selected and low when it is deselected.

STEP   is the inversion of the Amiga's STEPB* output. It is normally low but pulses high when any external drive is stepped.

If you build an external adaptor board, you should use a 74LS14 inverter to generate SEL and STEP from SELnB* and STEPB* (see Fig. 5). The 74LS14 has a Schmitt trigger input which gives better noise immunity in this application. If you modify your drive, you'll probably find a 74LS14 somewhere on the drive itself which has these signals on two of its outputs.

DISK IN   is a signal available somewhere on the disk drive PCB which goes high when a disk is inserted and low when it is removed. If you build an external adaptor board, it's convenient to wire this signal to Pin 2 of the 34 way drive connector (spare pin). This is the only modification that needs to be made to the disk drive if an external adaptor board is used.

MOTOR ON* is an input to the disk drive (34 way connector pin 16). MOTOR ON* is set low to start the drive's spindle motor and high to stop it.

IN USE*   is an input to the disk drive (34 way connector pin 4). IN USE* is set low to turn on the drive's front panel LED and high to turn it off.

If you build an external adaptor board, you will need separate MOTOR ON* and IN USE* lines going to each disk drive. All other inputs are common to all drives.

READY*   is an output from the drive which goes low when the disk is up to speed and ready to read from/write to. The Amiga uses it for this purpose as well as using it during power-up to identify the drive. Because READY* is driven by an open-collector output inside the drive, it is permissible for other open-collector outputs to be connected to it as is done in Fig. 1.

CHNG*   is an input to the Amiga which goes low if a disk has been removed from the currently selected drive since the last time that drive was stepped. If you build an external adaptor board, all CHNG* signals should be wired to Pin 11 of the Amiga DB23.

Whether you build an external adaptor board or modify your disk drives, you'll need a DB23 plug to connect to the Amiga's external drive port. Various club members have told me that they will be importing these by the bucketload real soon now, but if they haven't arrived by the time you need them, you can get by as I did by butchering a DB25. For this, you'll need a metal shell solder type DB25 plug. Extract pins 1 and 13 as shown in Fig. 3 and dent the ends of the shell (a delicate yet savage twist with a fine yet sturdy pair of pliers will do this nicely) so that the plug fits snugly in the port when turned upside down.

Extracting the pins will probably quite difficult unless you enlist the aid of your trusty soldering iron. Clamp the plug securely in a vice, then melt some solder into the connector end of pins 1 and 13 as if you were about to solder wires to them. Grasp the contact end with the aforementioned fine yet sturdy pliers, and push the pin gently but firmly into the connector while heating up the connector end with the soldering iron. You should find that the plastic holding the pin in place gives way with a minimum of mess.

I will not describe the making of an external adaptor board, because for various reasons (laziness, mainly; I *hate* Veroboard) I opted to modify the drives themselves. The Mitsubishi M4853 drives I used lent themselves quite nicely to these modifications. If you intend to do it the way I did, be aware that these mods will almost certainly void the manufacturer's warranty on the drives.

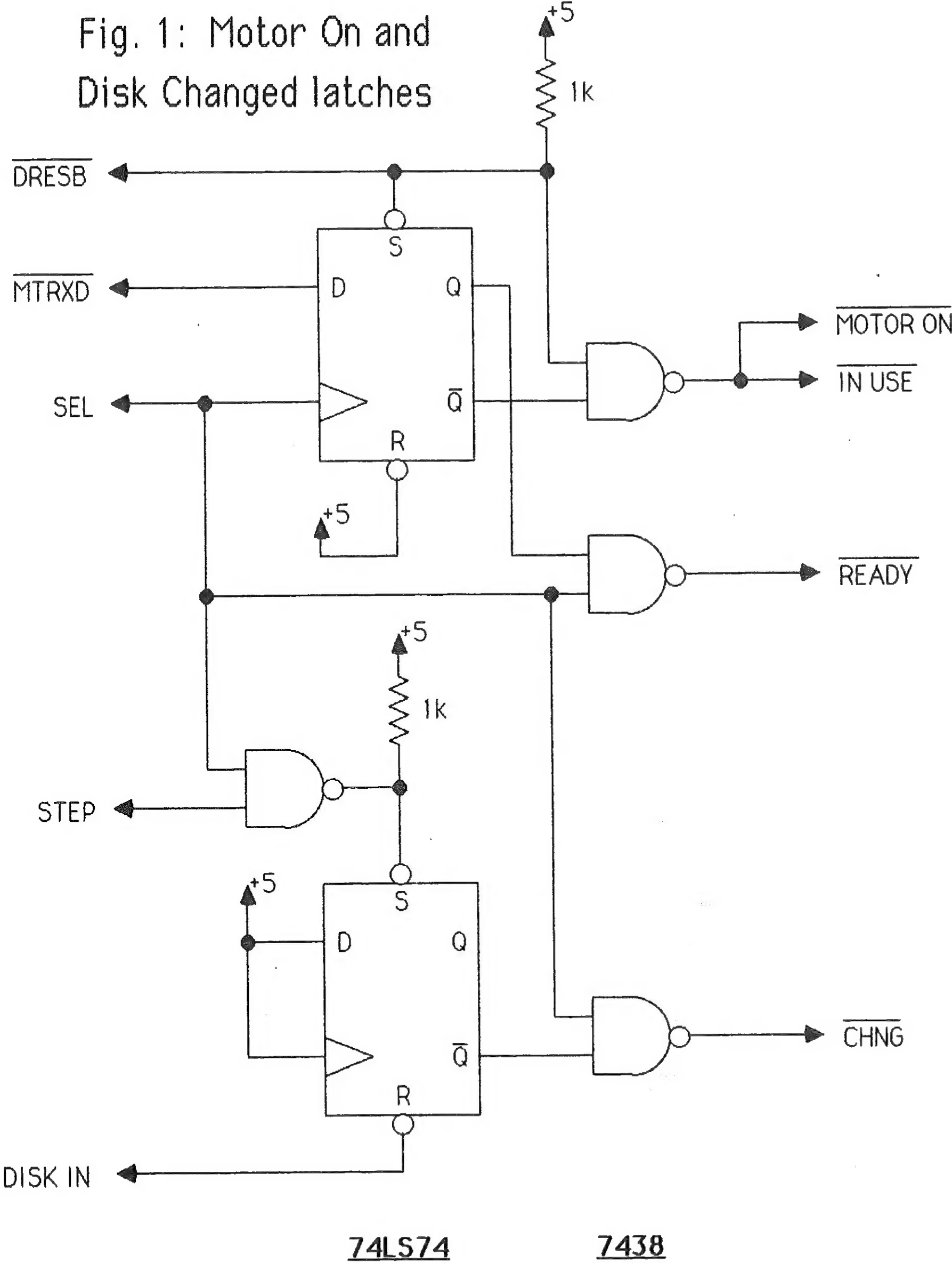## Fig. 1: Motor On and Disk Changed latches

DRESB

MTRXD

MOTOR ON
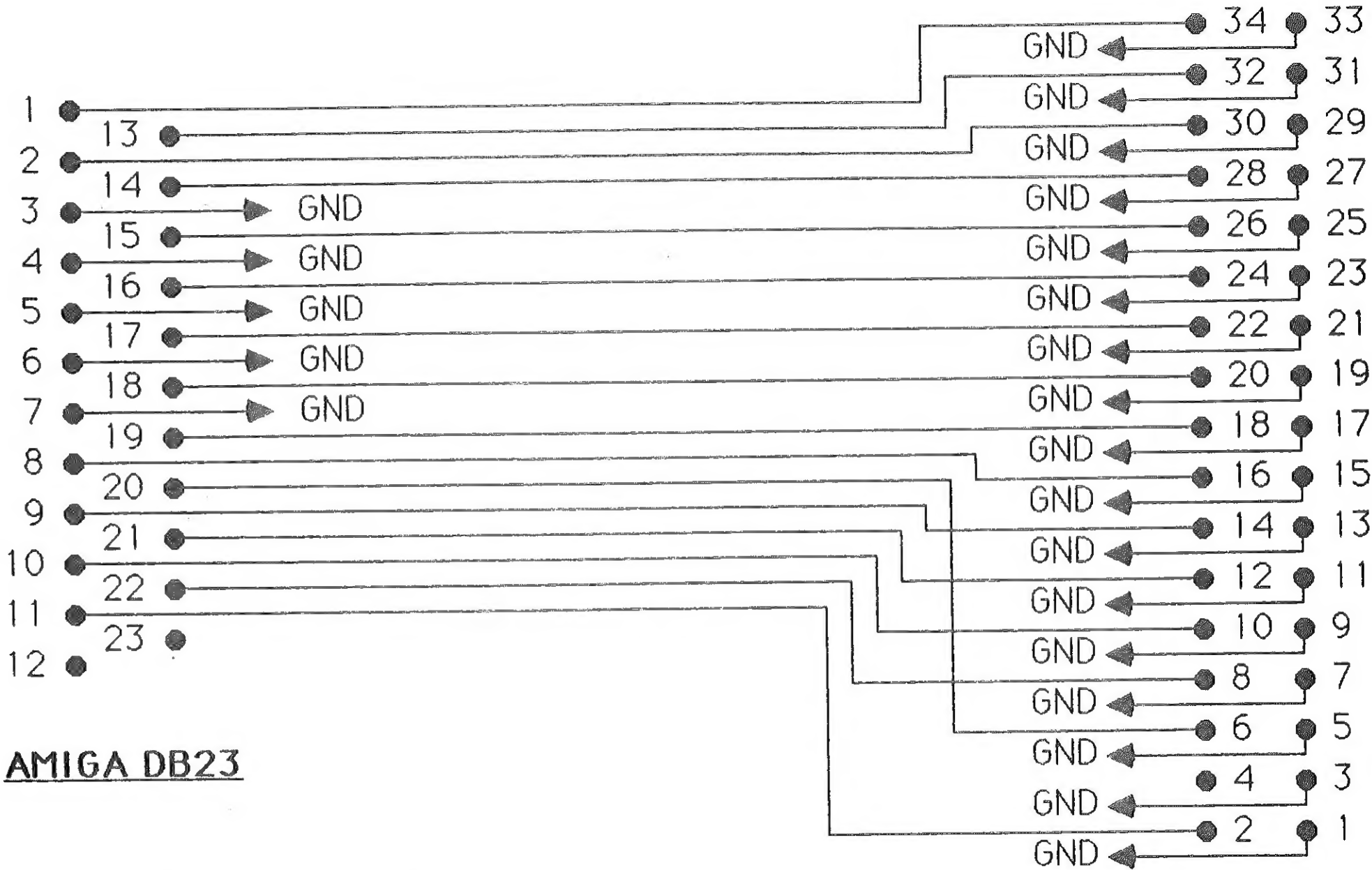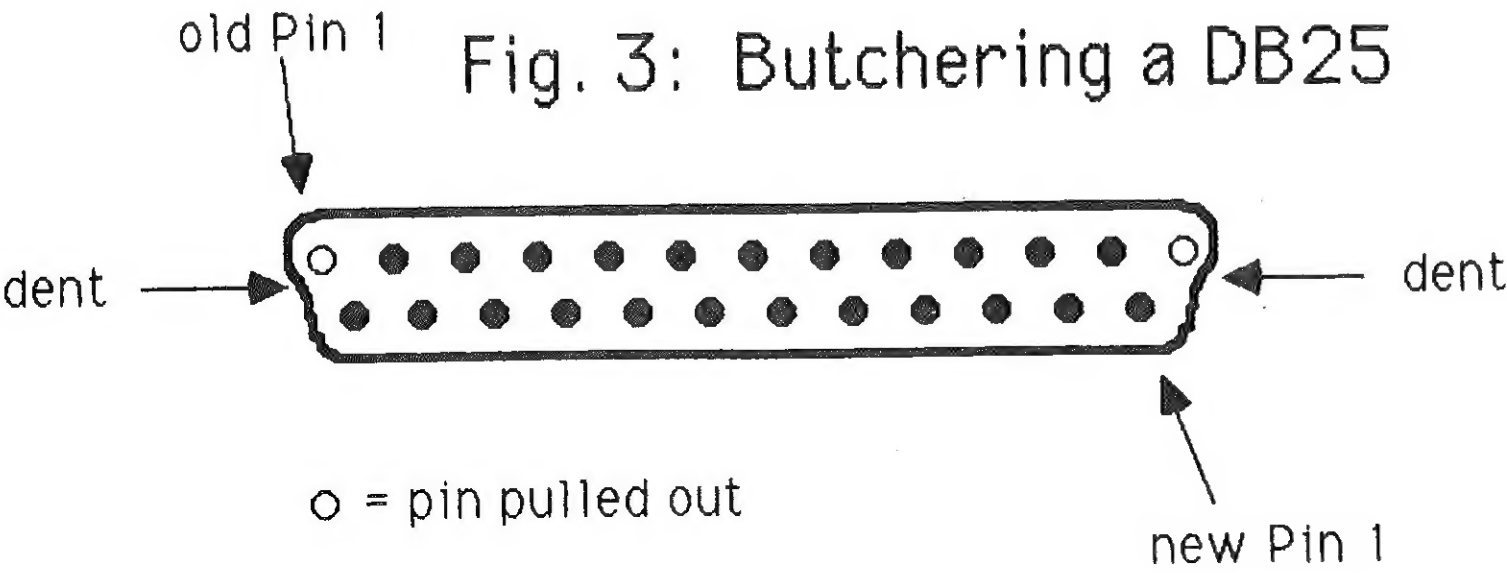
IN USE

SEL

READY

STEP

CHNG

DISK IN

74LS74   7438

Fig. 3: Butchering a DB25

old Pin 1

dent   dent

o = pin pulled out

new Pin 1

AMIGA DB23

DRIVE 34 WAY

Fig. 2: Cable for modified drives

DISK IN

MTRXD*
34-WAY
PIN 16

STEP
74LS14
PIN 10

SEL
74LS14
PIN 6

DRESB*
34-WAY
PIN 10

CHNG*
34-WAY
PIN 2

74LS74   7438

cut track

IN USE
34-WAY
PIN 4

Internal
MOTOR ON

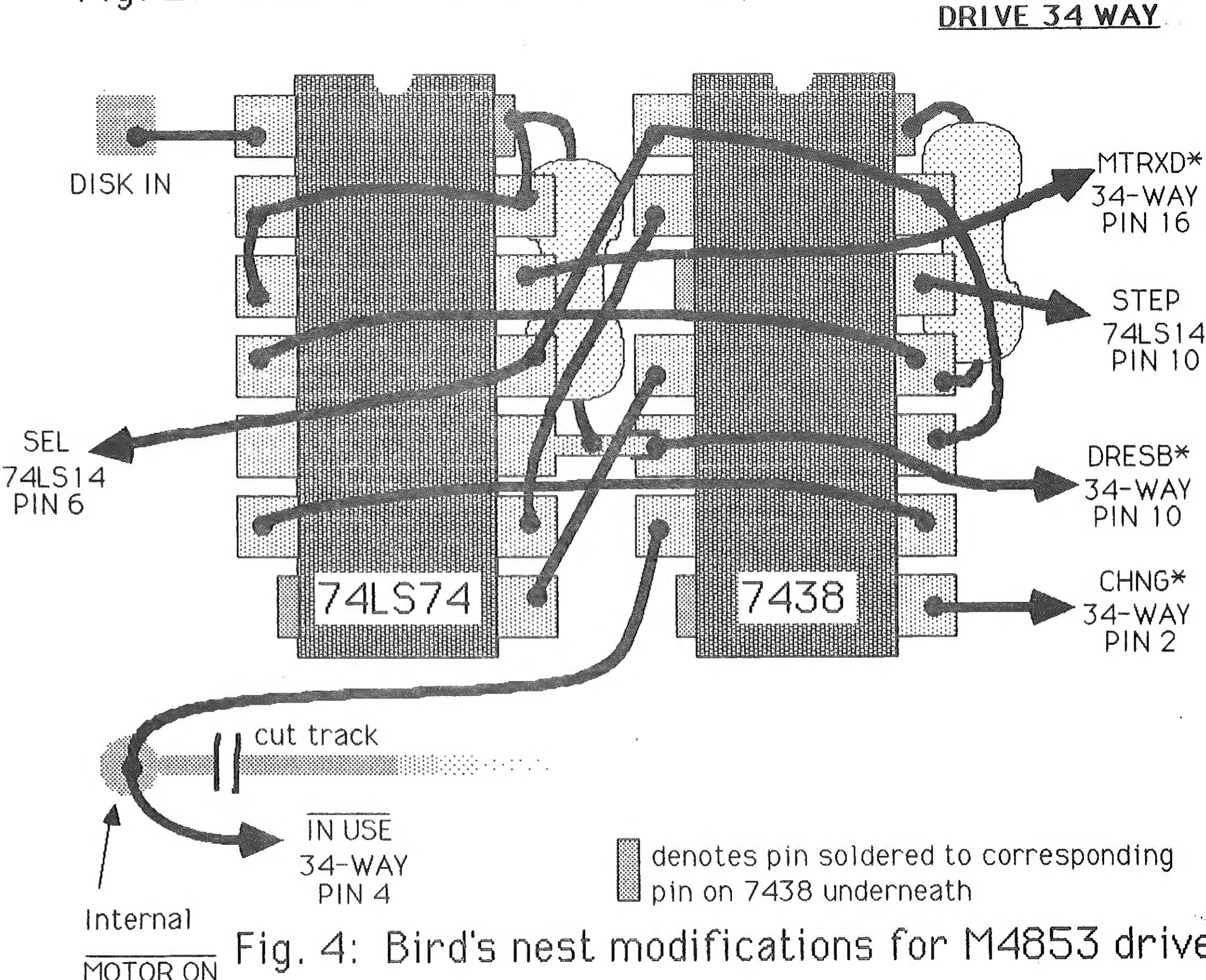denotes pin soldered to corresponding
pin on 7438 underneath

Fig. 4: Bird's nest modifications for M4853 drive

Fig. 4 shows the way I did mine. Near the 34 way interface connector on the M4853 drives is a pair of 7438 chips. The bird's nest in Fig. 4 is piggybacked on top of these. Note that most of the pins of the 74LS74 and the 7438 in the bird's nest are bent up and cut off, leaving a pad about 2 mm square for soldering onto. The exceptions are pins 3, 7 and 14 of the 7438 and pins 7 and 14 of the 74LS74, each of which is not bent up but soldered to the corresponding pin of the 7438 underneath, and pin 10 of the 74LS74. This last pin is bent up but not cut off, and reaches across to be soldered to pin 5 of the 7438. The 1K resistors should be mounted as shown. I found it easier to mount these to the ICs before soldering the latter down to the drive board.

Wire wrap wire should be used for all the interconnections, not standard hookup wire. This is a *very* small bird's nest, and hookup wire would interfere with the layout.

All references to pins on a 74LS14 refer to the one and only LS14 on the drive board, located mere millimetres away from the piggybacked chips. The square pad at the top left of Fig. 4 is a convenient place to pick up the DISK IN switch signal (verify it with a multimeter or logic probe before using it). The round pad labelled "internal MOTOR ON*" is the only through hole pad in the vicinity, so it's easy to spot. You should cut the track connected to this pad, instead of cutting the track connected to the MOTOR ON*
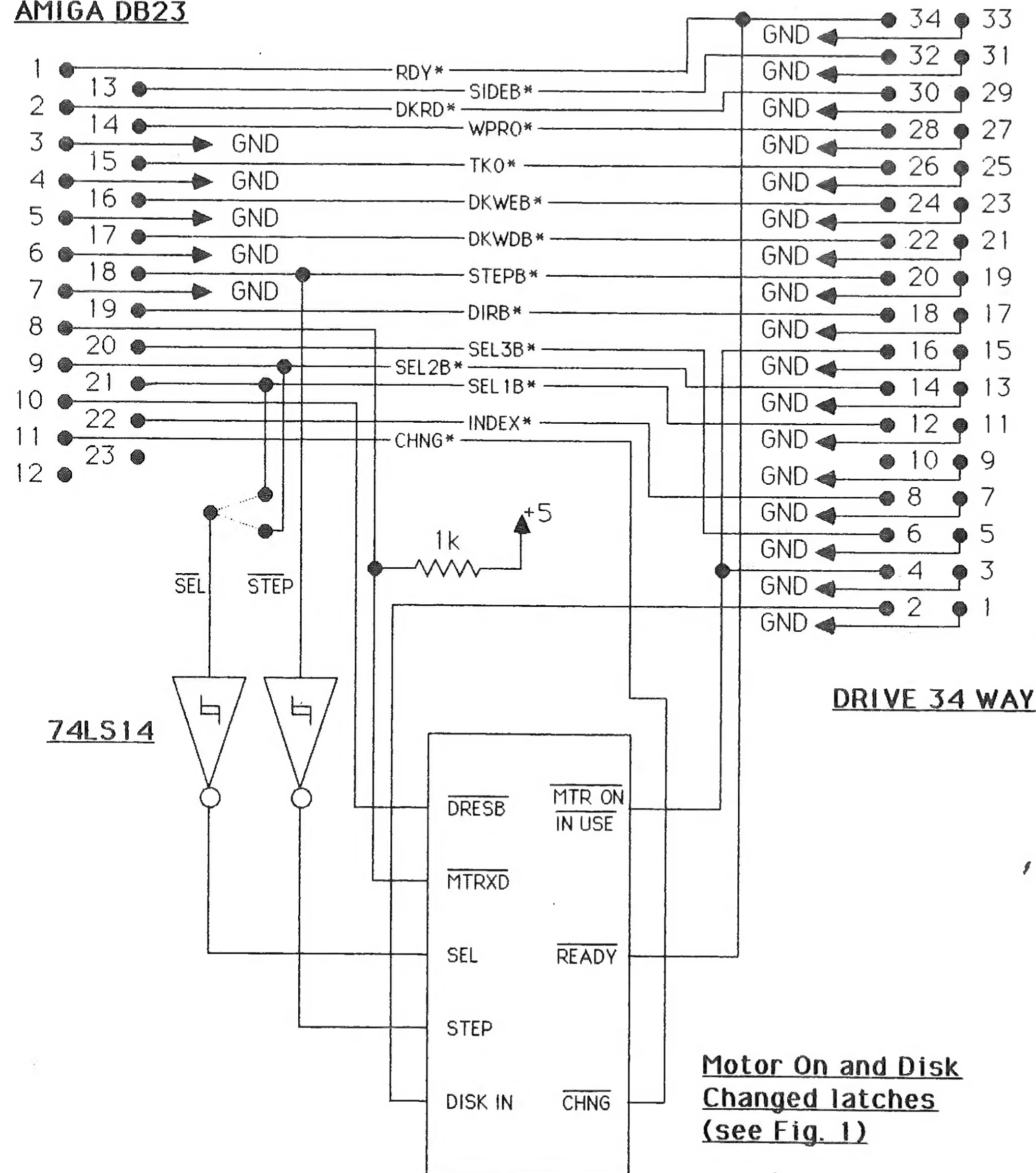
line on the 34 way connector (pin 16); this will allow the Amiga's MTRXD* to be pulled up by the resistor attached to that line. MOTOR ON* itself will be pulled up by the resistor on the IN USE* line once the two are connected together as shown.

If you use this scheme, be sure that nothing is connected to the IN USE* input of the drives, as shown in Fig. 2. If you have two external drives and the IN USE* lines end up being connected together, both motors and both activity lights will always go on together, rendering the whole latch idea useless.

On the subject of pullups: Rather than having a 150 ohm pullup pack in the last drive on the daisy chain, the Amiga drives use a 1K pullup pack in every drive. Those of you familiar with transmission line theory will recognize this as a bit of a compromise, to say the least; but it seems to work OK in practice provided cable lengths are kept fairly short. I ran mine for a while with the 150 ohm pullup in place, and the Amiga seemed to cope, but people I trust assure me that this was risking the lives of the line drivers in my machine, so I bit the bullet and fitted both external drives with 1K pullup packs.

If you elect to build an adaptor board as shown in Fig. 5, the only modifications you'll need on the drives is a wire between DISK IN (wherever it is hiding on the drive board) and Pin 2 of the 34 way connector. If you run more than one external drive, you'll need one set of Motor On and Disk Changed latches and one SEL inverter for each drive, plus one STEP inverter to all drives. All the lines that connect to the Amiga DB23 in Fig. 5 are common to all drives. The others are individual inputs or outputs of the latches.

I jumpered my drives to respond to selects 1 and 2. This results in them showing up as DF1: and DF2: if plugged directly into the Amiga, or as DF2: and DF3: if plugged into the daisy chain port on an external Amiga drive. If your drives have head load, jumper them for Head Load on Motor On.

There are various modifications you can make to these circuits. One worthwhile one would be to block the STEP* signal to a drive unless its motor is currently on, getting rid of the annoying clunk... clunk... clunk... that comes out of unoccupied drives as the Amiga polls them to see if there's a new disk in place. I'll leave the writeup of that mod to the first person keen enough to make it work.

Thanks to Drac, Lachlan, Neil, Peter, Robert and Ronnie for help and ideas, and to Maclagan, Wright & Associates for the use of the Macintosh and Imagewriter 2 used to draw the pretty pictures (which I will continue to use until somebody produces a package for the Amiga as useable and cost-effective as MacDraw).

- Stephen Thomas

AMIGA DB23

74LS14

DRIVE 34 WAY

Motor On and Disk Changed latches (see Fig. 1)

Fig. 5: Adaptor for minimally modified drive

Get "Wacked" by Eric Salter

Built into your Amiga's ROM kernel, is a debugger called `ROM-WACK`. A debugger is what computer programmers (usually system programmers/developers) use to see why their program failed and what happened to the machine. ROM-WACK sits there doing nothing until you activate it! Be warned however, it isn't much use to you unless you know what's going on and you have a serial terminal! You need to connect a serial terminal to the connector at the back, set at 9600 baud, 8 data, 1 stop, no parity.

To see what ROM-WACK does, you have to do one of two things:

CASE 1: If when you get a GURU MEDITATION number alert, push the RIGHT mouse button - not the left as asked, and from the serial terminal will eminate a stream of numbers including the program counter, stack pointer, register info and the TASK number that caused the failure. Now and again if the Amiga refuses to do something, like open a window, then occasionally the serial terminal will announce the fact!

CASE 2: When you load the Workbench Iconic Interface from the CLI with the command **Loadwb**, instead try **Loadwb -debug**, you'll get the familiar disk icon and menu bar, but if you slide the pointer along the bar while holding the menu select button down, you will come across an extra menu with the options of:

```
Debug
Variables
Wblist
Vollist
Flushlibs
```

All output is from the serial terminal. Debug gets you into ROM-WACK, Variables, lists the base rootobject, Wblist lists the available volumes and which is linked to the rootobject. Vollist lists the current assignment definition tables. There are several standard assignments that the operating system knows about. These are: the devices PRT:,PAR:,SER:, RAW:,CON:,RAM:,DF0:,DF1: etc... Devices are of type 0 and are not deletable or re-assignable. Type 1's which are deletable/re-assignable are things such as C:,FONTS:,S:,L:,DEVS:,LIBS:,SYS:. These are known about by the operating system and are different from the workbench sub-directories of the same name in that they happen to be assigned to them e.g.:

C: is assigned to Volume Workbench:c

What this means is that if the operating system is trying to interpret a command from the cli command line, it will search the current directory first i.e. the one that is listed with a `cd` command. If it fails to find it there, it will then search C: which it knows about and which is by default Workbench:c. Thus it looks at the c directory of workbench, and this is why if you have only one disk drive, and enter a command the system doesn't recognize, it will put up the requester ^Place Volume Workbench in any drive...." The practical upshot of this is that if you copy the c directory into RAM , then you can stop the requester going up by assigning C: to RAM:, makeing the operating system search RAM for the command.

Disk Errors

Have you ever noticed that one day, a disk develops a read/write errorwhere it hasn't before? This is to be expected occasionally, but it occurs with too frequent regularity on the Amiga! What's worse, is that when you try to reformat that disk from either the Workbench interface or the CLI, it will not and gives a validation/Bad block number error every time. Even if you do a "diskcopy" from either interface, the files that crashed previously, still take the computer to lunch with the Guru. If you want a solution to this problem then read on.

I cannot guarantee this will work for you but this is what happened to me after ages of absolute frustration. With the prospect of having to throw away an expensive disk, I had to try everything, as it turns out, this is virtually what you have to do! I made a back-up copy of my Extras disk, which is what everyone should do to all disks. This copy worked fine until I was using the Amiga Tutor to show off my new toy to a friend. As it turned out, the fault decided to occur just then. A system requester went up, telling me the volume had a read/write error. Needless to say, I had a visit from the Guru shortly afterwards. Hmm...I thought, the disk has an error, I can cope with that, I'll just reformat and she'll be right...wrong. Formatting served only to hasten on my apoplexy, the damn thing refused to verify what it had just formatted - real computers don't do this sort of thing, at least not to me. I then decided, the disk must have a fault in manufacture (this may yet turn up to be the case) so I took it back and got another which they found in the back room with junk already on it. In it went, and I tried to run the junk already there. Read/Write error!!! Hmm..."Is it my Amiga" I pondered, so I took out my trusty Dysan disk and all was well, whew! I then tried to re-format this disk and a veritable pandoras box of problems opened up. I got errors from validating formatted cylinders, CRC error messages, an icon that was labeled DFO:NDOS, which I can only assume stands for - yes this is not a Kick disk and it's Not DOS either,but there's something there - limbo! I tried a disk copy from both interfaces, all seemed well, but diskcopy doesn't verify what it's written, (quite dangerous) and trying to run a program brought on an attack of the galloping gurus, yes folks this was fun. Finally, in desperation and leaving out the re-tries, swearing, visit to the shrink, I did a diskcopy from workbench, then did an initialize from workbench and finally did a diskcopy from the CLI...result...no errors, no guru...It WORKED!!!

Why does this happen? To pack information onto an Amiga disk at the density it uses, there is a lot of housekeeping data that is left out of the format. This means that Amiga disk reads/writes are more critical that other formats and more subject to small errors that other operating systems could recover from. The other problem seems to be the way the formatting routines handle disks that are not blank. It seems that if a disk is blank, it is ok to format it (ie DFO:BAD), no problems, but if the disk contains DOS info, even if it is corrupted, it refuses do a damn thing about it and screams 'CRC error','Hard error' or 'error validating disk' et al, giving the partially initialized disk the name of 'DFO:NDOS'. This is pure speculation at the moment but I think it is what's happening. So if you give the format routine something to work with, everything is okay.

A Workaround for the HiRes bug

A meeting or two ago, I burst in very rudely and asked a question which I was politely informed had already been answered: What can I do to get rid of the nasty-looking spewglies that appear at the top of my screen when I display pictures in 640*400 mode? Neil Murray told me that one of the dealers had been having customers moan about this fault since the last shipment of Amigas came in and had traced the problem back to certain copies of the Workbench disk. He suggested that I contact them for a replacement.

Rather than trek all the way into the city for this boon (a big hassle for me since I lost my driver's licence), I persuaded a kindly fellow member to let me have a copy of his Workbench. He said he had never had any problem with the bug, so I assumed he was one of the lucky recipients of the fault free version.

Feeling confident that I had the problem tamed at last, I took home his copy, used Preferences to set it up for my peculiar taste in video displays, stuck Soft Demos in the drive, double clicked on the Mandril... and there he was, gazing out from the screen with his usual bloodshot, flickering expression, and a generous helping of prime HiRes spewglie floating above his forehead.

After a short pause to let my blood pressure settle back to normal, I started to think about the problem in a rational, scientific manner, and came to the conclusion that the best thing to do would be to twiddle all the knobs and press all the buttons until the problem went away.

Here's how you get rid of the gunge:

Get into Preferences. Grab the little right-angle gadget you use to centre the display on your monitor and drag it all the way down to the bottom. You should see the bottom border of the window disappear. Now move the display up slowly, counting the number of one-scan-line jumps, until you have moved it up at least six lines (this will involve moving the right-angle gadget up about twelve lines). Click on the SAVE or USE button, then go and display the Mandril. You should find that the spewglies have disappeared.

It seems that if you try to put the display anywhere in the bottom six possible positions, your machine will have trouble displaying 640*400 images. I don't now why this.is. If there is anybody out there who CAN display the Mandril properly no matter where your screen is set, I'd like to hear from you - please ring me (Stephen Thomas) at home on 836-7364 (decent hours only please).

Information Page

AMIGA Users Group

The **AMIGA Users Group** is a non-profit, self-help group, consisting of people interested in the Commodore AMIGA computer and related topics.

Club Meetings

The **Amiga Users Group** meets at 2pm on the second Sunday of each month at:

Victoria College, Burwood Campus, in the Community Resources Centre (Building E) (Melways map reference 61 B5)

Coming Meetings are: **August 10th & September 14th**

Production Credits

This month's **AMIGA WORKBENCH** was edited by Peter Jetson. Equipment and software used was: TurboDOS S-100 computer, Diablo 630 printer, Gemini 10x printer, Wordstar and Fancy Font.

Copyright and Reprint Privileges

Articles herein that are copyrighted by individual authors or otherwise explicitly marked as having restricted reproduction rights may not be reprinted or copied without written permission from the **AMIGA Users Group** or the authors. All other articles may be reprinted for any non-commercial purpose if accompanied by a credit line including the original author's name and the words "Reprinted from **AMIGA WORKBENCH**, newsletter of the **AMIGA Users Group**, PO Box 109, North Balwyn, 3104".

Contributions

Articles, papers, letters, drawings and cartoons are actively sought for publication in **AMIGA WORKBENCH**. It would be appreciated if contributions were submitted on disk, since that means they don't have to be re-typed! We have access to a wide range of computers, so we should be able to accept almost any type of disk, but AMIGA disks are certainly the easiest. Absolute deadline for articles is the last weekend of the month before the cover date. Contributions can be sent to:

The Editor, AUG, PO Box 109, North Balwyn, 3104

AUG Users Group Disks

Disks from the **AMIGA Users Group** Library are available on quality 3.5" disks for $10 each including postage. Currently, the group holds 25 public domain volumes, sourced from the USA.

Member's Discounts

The **AMIGA Users Group** is currently negotiating discounts for its members on hardware, software and books. Members will be notified when negotiations are complete.

Currently, **Technical Books** in Swanston Street in the city offers **AUG** members a 10% discount on computer related books. Just show your membership card.

Membership and Subscriptions

Membership of the **AMIGA Users Group** is available for an annual fee of $20. To become a member of AUG, fill in the form on this page, and send it with a cheque for $20 to:

AMIGA Users Group, PO Box 109, North Balwyn, 3104

Application for membership of The Amiga Users Group

Membership is $20 per year. Make cheques payable to The Amiga Users Group, and send to:

Amiga Users Group
PO Box 109,
North Balwyn, 3104

Details this side are optional

Surname: _____

First name: _____ (no initials)

Address: _____

_____ Postcode: _____

Where did you hear about AUG: _____

How can AUG help you: _____

_____

_____

Signed: _____ Date: _____

Phone number: _____ STD code: _____

Year of birth: _____

Occupation: _____

Interests: _____

_____

_____

Dealer's Name: _____

Dealer's Address: _____

_____ Postcode: _____

Are you happy with your dealer: _____

In the event of my admission as a member, I agree to abide by the rules of the Association for the time being in force.